ONLINE DIAGNOSING OF COMPUTER HARDWARE AND SOFTWARE

FIELD OF THE INVENTION

The present invention relates to the diagnosing of computers from a remote location. In particular, the present invention relates to a method and apparatus for diagnosing computer hardware, software and performance problems from a remote location, at least substantially without requiring human assistance at the remote location.

BACKGROUND OF THE INVENTION

The diagnosis and remediation of problems associated with the operation of computers is increasingly important to the computer industry. In particular, assisting both experienced and inexperienced computer users in remedying recurring problems has become a significant drain on computer hardware and software manufacturers. Typically, problems encountered by computer users are dealt with by human telephone operators. The telephone operators, or "support personnel," may or may not have access via a communications link to the problem computer. Generally, the process of diagnosing and remedying computer problems from a remote location is time consuming for both the computer user and the support personnel. Furthermore, the provision of support personnel is a significant cost to the computer industry.

In order to reduce the expenses associated with maintaining a cadre of support personnel, the computer industry has implemented various approaches. For instance, some software and hardware companies severely limit user access to support personnel. Access may be limited to recent purchasers of a product, or the user may be charged a fee for contacting the support personnel. However, limiting the access of customers to assistance by restricting those customers who are eligible for such assistance or by charging fees for such assistance generates

ill will among customers. In addition, such approaches do not reduce the time required to successfully remedy problems and they do not reduce the difficulty of diagnosing and remedying problems from a remote location.

As an alternative or in addition to restrictions on access to support personnel, some computer hardware and software providers supply "online" help. Such online help may be in the form of statements of typical problems and suggested solutions to those problems. This online help may be stored in media provided with the hardware or software or may be accessible to the user by contacting a specified server over a computer network, such as the Internet. However, such systems are ineffective if the computer user is unable to recognize his or her particular problem from among the included problems and suggested solutions. In addition, such systems provide assistance only with problems directly associated with a particular piece of hardware or software. Therefore, they are generally unable to assist a user in successfully addressing, for example, problems that affect the particular hardware or software but that are caused by another hardware or software component in the user's computer.

The providers of computer hardware and software have also utilized programs that allow support personnel at a remote location to operate or obtain information from a user's computer from the remote location. Such systems have the advantage of allowing trained personnel to directly address the user's computer. However, users may object to allowing a stranger to have complete access to the user's computer. In addition, such systems do not remove the need for maintaining support staff, and such systems have a relatively high impact on the resources of the user's computer.

Still another approach to providing support for computer users is to provide automated or semiautomated diagnostic programs. Such programs may be resident on the user's computer, or may be accessed over a computer network, such as the Internet. However, providing a support program resident on a user's computer having the ability to diagnose and potentially address a wide range of problems requires a relatively large program. Particularly where a user's computer is not operating properly, loading or accessing such a program can be problematic. Similarly, diagnostic programs that allow a user to interconnect to them over a computer network generally have a high impact on the user's computer, in that the programs are relatively large and require a relatively large amount of memory and resources to operate. For instance, Internet browsers may allow access to a computer's system information and control settings, however the browser components necessary to allow such interactivity, such as ACTIVE X or other browser "plug ins," demand a relatively large amount of resources on the user's computer. In addition, not every type or version of browser is compatible with a particular browser plug-in, and the user may be required to obtain and install additional or alternative browsers or plug-ins. Accordingly, the operation of such systems can be problematic. Furthermore, such systems are incapable of effecting automatic repairs to a user's computer.

It would be advantageous to provide a method and apparatus that enabled computer problems to be diagnosed and remedied from a remote location. In particular, it would be advantageous to provide a method and apparatus to remotely diagnose and remedy computer problems from a remote location that essentially did not require involvement by support personnel at the remote location. In addition, it would be advantageous to provide a method and apparatus for diagnosing computer problems from a remote location capable of operating on

computers having limited resources available as a result of a problem or problems requiring diagnosis. It would also be advantageous to provide a method and apparatus that did not require a user to state the exact nature of the problem in order to obtain a suggested solution. Furthermore, it would be advantageous to provide a method and apparatus for diagnosing computer problems from a remote location capable of successfully diagnosing a majority of presented problems, and that could be provided at reasonable cost.

SUMMARY OF THE INVENTION

In accordance with the present invention, a method and apparatus for diagnosing computer problems from a remote location are provided. The present invention generally allows diagnostic tests or individual commands to be downloaded from a remote location and operated on a computer or other device to be diagnosed. In particular, the present invention allows for the diagnosis of a computer without requiring a large amount of the resources of the computer to be diagnosed. Furthermore, where an automated solution is available, the present invention is capable of repairing the computer.

In accordance with one embodiment of the present invention, method and apparatus are provided that allows for the remote diagnosis of computer problems. According to the method and apparatus, the user of a computer to be diagnosed initiates contact with a diagnosing server. This contact may be made over a computer network, including a wireless network, using a communications interface, such as an Internet browser program. A client application is then downloaded from the server to the computer. The client application is capable of executing scripts or individual commands that may be downloaded from the diagnosing server. In general, the scripts comprise diagnostic tools for collecting various information regarding the client

computer. The results obtained from executing the client scripts or individual commands are returned to the diagnosing server. The returned results are compared to stored results using a rules based analysis, and a disposition is returned to the browser of the computer. The disposition may include automatically performing a maintenance or repair procedure, may suggest that the user perform a particular maintenance or updating procedure, may automatically take the user to a web page or information source, or may refer the user to a source of further information or assistance regarding the diagnosed problem.

In accordance with another embodiment of the present invention, a client application running on a computer to be diagnosed is provided with an identifier generated by the diagnosing server by passing the identifier from the title bar of the browser to the client application.

According to still another embodiment of the present invention, the client application polls a server application running on a remotely located server. In particular, the client application polls the server for instructions concerning a next operation to be performed in connection with the computer. For instance, the client application may poll a server application for an instruction to execute a script or command. After executing a script or command, the client application signals the server application that execution has been completed. An indication that execution of a particular script has been completed is entered in a database record associated with the client computer and identified by the identifier. In addition, the browser may poll the server application to detect an indication that execution of the script has been completed. In response to detecting such an indication, the browser or screen displays a next page to the user.

According to yet another embodiment of the present invention, a user establishes contact with a remotely located server using an Internet browser or other communications interface. The

user registers with the server and is assigned an identifier. A client application is downloaded from the server and installed on the client computer or device. Next, a script or command, for execution using the client application, is downloaded to the client computer. The client application polls a record in a database maintained by the remotely located server and having the same identifier as the client computer to detect a signal to begin execution of the script or command. The indication to execute the script or command may be provided in response to a signal from the browser or other communications interface authorizing such execution. Upon detection of the indication, the script is executed, and the results are returned to the server. The browser or other communications interface polls the database record to detect an indication that execution of the script or command has been completed. Upon detecting such an indication, the browser or other communications interface displays a next page.

According to still another embodiment of the present invention, the user of the client computer is prompted to specify whether the problem is believed to be a hardware, software or performance problem. Based on the user's selection, additional scripts or commands may be downloaded to the client application and executed. For instance, if a user specifies a software problem, a script may be downloaded to the client application to determine the nature of the problem. For instance, upon execution, such a script can instruct the user to start the problem application. The script may then determine the amount of time required for the computer to load the application, or may collect an error message generated in response to the user's attempt to start the problem application and return that error message to the server for analysis.

A variety of diagnostic tools in the form of scripts or commands may be downloaded to and executed by the client application. In general, the scripts or commands are provided to

perform inventory and diagnostic functions. The client application and associated scripts or commands are designed to have a low impact on the system resources of the computer.

Based on the foregoing summary, a number of salient features of the present invention are readily discerned. A method and apparatus for diagnosing problems on a computer or other device from a remote location are provided. The method and apparatus of the present invention enable problems to be diagnosed and potential solutions proposed to a user of the computer or other device substantially without requiring human intervention at the server side. The present invention may also provide an automated solution to a problem on a computer being diagnosed. In addition, the method and apparatus of the present invention have a low impact on the resources of the computer or other device being diagnosed, thereby increasing the probability that the method and apparatus will successfully diagnose problems even on computers or other devices having reduced system capabilities due to the problem or problems requiring diagnosis.

Additional advantages of the present invention will become readily apparent from the following discussion, particularly when taken together with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram depicting the major components of a system in accordance with an embodiment of the present invention;

Fig. 2 is a block diagram depicting a server in accordance with an embodiment of the present invention;

Fig. 3 is a block diagram depicting a client computer in accordance with an embodiment of the present invention;

Fig. 4 is a block diagram depicting the logical lines of communication between a server and a client computer in accordance with an embodiment of the present invention;

Fig. 5 is a flow chart illustrating the operation of a system in accordance with an embodiment of the present invention;

Fig. 6 is a diagram depicting the operation of a client application in accordance with an embodiment of the present invention;

Fig. 7 is a diagram depicting the operation of a communications interface in accordance with an embodiment of the present invention;

Fig. 8 is a flow chart depicting the operation of test modules in accordance with an embodiment of the present invention;

Fig. 9 is a flow chart depicting the operation of a system in accordance with an embodiment of the present invention in response to the selection of a hardware problem;

Fig. 10 is a flow chart depicting the operation of a system in accordance with an embodiment of the present invention in response to the selection of a software problem;

Fig. 11 is a flow chart depicting the operation of a system in accordance with an embodiment of the present invention in response to the selection of a performance problem;

Fig. 12 is a flow chart depicting the operation of a system in accordance with an embodiment of the present invention in response to the selection of an unknown problem; and

Fig. 13 is a flow chart depicting the provision of a disposition to a client computer in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

Fig. 1 illustrates a system **100** in accordance with an embodiment of the present invention. The system **100** generally includes a server **104** interconnected to a plurality of client computers **108**a to **108**n by a network **112**. In general, the network **112** may be any computer network or means of establishing communications between two computers, including the Internet and wireless methods.

With reference now to **Fig. 2**, a server **104** in accordance with an embodiment of the present invention is illustrated. The server **104** generally includes a processor **200**, a network interface **204** and storage **208**. The storage **208** may include a database **212** and stored programs **216**. The stored programs **216** may include a client executable package **220** and a server application program **224**.

In general, the processor **200** may be any computer processor suitable for use in a general purpose server computer. For instance, the processor **200** may comprise one or more INTEL PENTIUM® class microprocessors. The network interface **204** may be any communications interface suitable for interconnecting a server computer to a communication network **112**. For example, the network interface **204** may comprise a TCP/IP interface or others. The storage **208** may comprise any mass media storage device, such as a hard drive, tape drive, or optical drive. In addition, the storage **208** may comprise multiple devices of various types.

The database **212** may be any control program and associated records suitable for storing and organizing information. The client executable package **220** generally includes programs for downloading to client computers **108**. The server application program **224** includes the commands and instructions required to coordinate the registration of users with the system **100**,

the downloading and execution of components of the client executable package **220**, and the exchange of information between the database **212** and the client computers **108**. Accordingly, the server application program **224** may include components to interface the database **212** to the client computers **108**.

Fig. 3 depicts a client computer **108** in accordance with an embodiment of the present invention. The client computer **108** may comprise a central processing unit (CPU) **300**, input/output devices **304** and peripheral devices **308**. According to one embodiment of the present invention, the client computer **108** is any general or personal type computer. According to a further embodiment of the present invention, the client computer **108** is a PC computer running a MICROSOFT WINDOWS® operating system. As used herein, the term client computer may, in addition to a general or personal type computer, also refer to a variety of digital devices capable of communicating with another digital device, including personal digital assistants (PDA's) or other hand held devices.

According to one embodiment of the present invention, the CPU **300** generally includes a processor **312**, storage **316**, a network interface **320**, a video card **324** and a plurality of internal devices **328**. The internal devices may comprise memory **332**, a hard disk drive **336**, a floppy disk drive **340**, a CD-ROM drive **344**, a tape drive **348** and other internal devices **350**. The input/output devices may include a video display or monitor **352**, a keyboard **356**, a mouse **360** and other input/output devices **364** such as audio devices and additional pointing devices. The peripheral devices **308** may generally include a printer **368**, a scanner **372** and other external devices **376**. The client computer **108** may also include operating software **380** and application

software **384**. In general, the various components of the client computer communicate with one another over a communications bus **388.**

With reference now to **Fig. 4**, the logical lines of communication between the server **104** and the client computer **108** are illustrated. In general, a first logical line of communication **408** extends between the server application **224** and a communications interface **404**. The communications interface **404** may include any communications application running on the client computer or device **108** that is capable of passing data between the client computer or device **108** and the server application program **224**. For example, the communications interface may comprise an Internet browser. Similarly, a second logical line of communication **412** extends between the server application **224** and the client application **400**. A third logical line of communication **416** extends between the client executable package **220** and the client application **400**. As is evident from **Fig. 4**, except for an identifier communication path **418**, there is generally no direct line of communication between the client application **400** and the communications interface **404**. Accordingly, with the exception of passing an identifier from the communications interface **404** to the client application **400** over the identifier communication path **418**, as will be described below, the client application **400** and the communications interface **404** communicate via the server application **224**. The physical communications link over which the logical communication channels **408, 412** and **416** are established may comprise the computer network **112** and associated network interfaces **204** and **320** described above. The physical communications link over which the identifier communication path **418** is established may comprise the communications bus **388** or internal registers.

The operation of the client application **400** and the communications interface **404** are synchronized to each other by monitoring a field in tables or records **420** stored in the database **212** through an interface provided by the server application **224**, as will be explained in detail below. This arrangement allows the client application **400** to be implemented as a relatively small program that requires relatively few of the client computer's **108** system resources to operate. In particular, it allows the client application **400** to control aspects of the operation of the client computer **108** without requiring a program that is integrated with the communications interface **404**. More particularly, the client application **400** is selectively provided with diagnostic tools, such as individual commands or application scripts **424** that perform selected functions. For example, the diagnostic tools may comprise code that is capable of execution while residing on the client computer **108**. This is advantageous, as programs integrated with a communications interface **404**, such as an Internet browser, that are capable of detailed interaction with the operational aspects of the client computer **108** typically consume a relatively large amount of system resources. In addition, such programs may not be compatible with all versions or types of a communications interface **404**. Accordingly, the arrangement illustrated in **Fig. 4** enables the present invention to function even on client computers **108** having compromised performance characteristics, for example due to a problem requiring diagnosis. The arrangement illustrated in **Fig. 4** also allows the present invention to operate in cooperation with a wide range of communications interfaces **404**.

With reference now to **Fig. 5**, the operation of a system **100** in accordance with an embodiment of the present invention is illustrated. Initially, the user of a client computer or device **108** establishes a connection with the server **104** (step **500**). The user may establish the

connection by, for example, directing a communications interface **404** on the client computer **108** to a website providing access to the server **104** over a computer network **112**. According to one embodiment of the present invention, active server pages may be used to provide the user of the client computer or device **108** with information from the server application **224**. Information from the user may be provided to the server application **224** using the communications interface **404**.

As part of establishing a connection with the server **104**, the user may be required to register with the server **104**. Registration may include the provision of information identifying the user to the server **104**. In particular, registration may include the creation of a record **420** in the database **212** for recording various information collected from the user. For instance, the record **420** for user information may include fields for a user name, a user address, and billing information, such as a credit card number and expiration date. Furthermore, the record **420** for user information may include a field for storing an identifier assigned by the server application program **224** and used by the system **100** to distinguish a particular client computer **108** from any other client computers **108** that are or have been in communication with the server **104**. The identifier also allows an individual user to register and diagnose more than one client computer **108**. A field for indicating a next script or command and a next page of information to display may also be provided. In addition, a script status field may be provided and used to synchronize the operation of the client application **400** and the communications interface **404**.

Following the completion of registration, the client application **400** is downloaded to and installed in the client computer **108** (step **504**). A first command, script or set of scripts **424** may be downloaded with the client application **400**. As used herein, the term script **424** should be

understood to include a command or set of commands capable of being executed by the client application program **400**. The command or commands comprising a script **424** may define a diagnostic tool that is useful in triggering execution of a program code that is already resident in the client computer **108**. In general, the client application program **400** receives scripts **424** from the server **104** and executes those scripts **424**. According to one embodiment of the present invention, a number of scripts **424** are provided. The use of a number of relatively small scripts **424** minimizes the impact that running the scripts **424** using the client application **400** has on the client computer **108**. In addition, the use of multiple scripts **424** allows the diagnosis of the client computer **108** to progress at a steady pace, rather than requiring a relatively long initial wait to download program code before testing can begin. Furthermore, the use of multiple scripts **424** allows the download and performance of only those tests that may be useful in diagnosing a particular client computer **108**, thereby providing an efficient test process.

The scripts **424** may be grouped as one or more test modules or diagnostic tools that perform various functions. The diagnostic tools may themselves comprise one or more scripts **424**, or they may comprise tools already resident in the client computer that are launched by a script **424**. In general, each of the scripts **424** performs a discrete operation or set of operations when executed by the client application **400**, as will be explained in the discussion set forth below.

At step **508**, the various tests are performed on the client computer **108**. According to one embodiment of the present invention the various test modules and other program components are executed immediately or almost immediately after they are downloaded to the client computer **108**. Therefore, according to this embodiment of the present invention, the step of downloading program components (step **504**) may alternate with the step of executing the test

modules or commands. The results obtained from executing the test modules and any other program components are then returned from the client application **400** to the server application **224** over logical line of communication **412**. The return of collected information to the server application **224** may also be carried out following the execution of each test module or individual test script **424**.

According to one embodiment of the present invention, general tests may be performed, and additional tests may be performed after responses to questions presented to the user have been elicited. Accordingly, the present invention may provide an interactive problem solving method.

The server application **224** stores the returned results in a record or records **420** maintained on the database **212**. According to one embodiment of the present invention, a record **420** is provided for each major category of information relating to the client computer **108**. For example, the records **420** may include individual records for information concerning the user, monitor, ports, printers, storage devices, system settings, video adaptor, performance, software, audio devices, component manufacturers and disposition. Additional or fewer types of records may be used. In general, the records **420** containing information concerning a particular client computer or device **108** are identified by the identifier assigned to that client computer **108** by the server application **224** when the user registered with the server application **224**. In addition, the records **420** may include records containing general information concerning particular problems and associated dispositions, including scripts **424** to perform procedures, including repair procedures, information regarding potential fixes, and contact information for automated transport to sources of additional support regarding particular devices or problems. Where

automated transport is provided, the communications interface **404** may be automatically directed to a Web page, such as a Web page maintained by the manufacturer of a problematic device or program, that contains relevant information. The database **212** may also store the scripts **424** to perform maintenance procedures and software patches, or links to such scripts **424** and software patches.

At step **512**, a rules-based determination of the disposition of the client computer **108** is conducted. In general, information obtained from the execution of the test modules or commands and other software components is analyzed, and a disposition concerning the client computer **108** is arrived at. The rules-based analysis is, according to one embodiment of the invention, adaptive in that the disposition returned to a client computer **108** in response to a particular problem may change as additional or alternative dispositions are added to the database. For example, the disposition may include a suggestion to install a software patch, to run a maintenance procedure, to perform a repair procedure, to perform a specific further test (*e.g.*, to check that power or signal cables are properly connected), to change a system setting, or to obtain additional information from another source.

According to one embodiment of the present invention, the rules based analysis of the disposition of the client computer **108** includes storing information returned from the execution of diagnostic tools including scripts **424** in the database **212**. In particular, the information returned to the server **104** is stored in various records **420** maintained or created in the database **212**. In general, the information contained in a record **420** may be limited to information concerning a particular device or type of device associated with the client computer **108**. In particular, information regarding components, devices or software associated with a client

computer **108** may be stored in database records **420** that contain information directed to a particular component, device or software program associated with the client computer **108**. Where the user has specified a component, device or software program as problematic, the record **420** containing information concerning the identified item may be queried to determine whether the item successfully passed earlier diagnostic tests. For instance, the record **420** may be queried to determine whether the item passed or failed the earlier test. A new record **420** containing disposition information may then be created, to store information related to the item. Next, a table in the database **212** containing support information may be accessed by the server application **224**. In general, the table containing support information comprises records **420** pertaining to problems and solutions for various components, devices or software associated with client computers **108**. In general, a disposition stored in a record included in the disposition table directed to the same problem detected or identified in connection with the particular client computer **108** being diagnosed is returned to the client computer **108**. The particular disposition returned may vary depending on whether the component, device or software program was found to have passed or to have failed earlier tests. Furthermore, it should be understood that the particular disposition may include a plurality of suggested dispositions.

The particular disposition returned initially to the client computer **108** is, according to one embodiment of the present invention, the solution that has been determined to be the most likely one to remedy the identified problem. If the initial disposition is not successful in remedying the identified problem, additional dispositions may be provided. Furthermore, the disposition returned to a client computer **108** may change from one session to another as information regarding solutions that are successful in remedying particular problems is gathered.

The disposition is then returned to the client computer **108** (step **516**). Where the disposition includes a suggested maintenance procedure, repair procedure, software fix or change in system settings, that procedure may be carried out automatically. For example, the return of a disposition to the client computer **108** may include providing a script **424** for execution by the client application **400** that performs the suggested maintenance operation, repair procedure, software fix or change in system setting automatically. The operation may be an operation that can be carried out using a software application already installed on the client computer **108**, in which case the script **424** simply activates that application. Alternatively, the script **424** may itself include an executable program or command for carrying out the procedure.

If the suggested remedial procedure involves a software fix, the step of returning a disposition to the client computer **108** may include providing the necessary software patch or software program. For example, if a device driver has been found to be faulty, a software patch or new driver may be downloaded to the client computer **108** and installed automatically. Additionally, if the error message indicates a missing software component such as .dll, .exe or call, the software component can be replaced or updated. In the case of a system setting that requires modification, the return of a disposition to the client computer **108** may include the provision of a script **424** that, when executed by the client application **400**, is capable of making the required change to the system settings.

In the examples given above, implementation of the suggested fix may be automatic in that the user of the client computer **108** is not required to search for the maintenance procedure, software patch or system switch, and then implement the fix. Rather, the server application **224**, as part of the disposition, provides a script **424** and any additional software necessary to perform

the procedure. According to one embodiment of the present invention, the user of the client computer **108** is asked to authorize implementation of a suggested remedial procedure before it is carried out, to allow the user to control the changes that are made to the client computer **108**.

In other instances, the disposition may include providing the user of a client computer **108** with a suggested solution to a diagnosed problem, but may require the user to implement the solution. For instance, a peripheral device **308** associated with the client computer **108** may require a new driver. However, that driver may not be available to the server application **224** for downloading to the client computer **108**. In such instances, the disposition may include automatically directing the communications interface **404** to a website or universal resource locator (URL) address from which the user may be able to obtain the necessary driver. In other instances where the automatic solution of a problem is not possible, the disposition may provide the user with information regarding how to perform a required maintenance operation, or may provide contact information. For example, information regarding a maintenance operation may include instructions to the user on how to use a software utility, or on how to check a physical data or power connection between the CPU **300** and a peripheral device **308**. Contact information may include a website address or a telephone number to allow the user to obtain the necessary information, software, or other fix.

In addition, a disposition may include a combination of automatic and manual fixes to problems. For instance, if a peripheral device **308** on the client computer **108** requires a driver that is not directly available to the server application **224**, the communications interface **404** may be automatically sent to a website address from which the required driver may be available. The user may then obtain the required driver from the website. According to one embodiment of the

present invention, a script **424** is then executed by the client application **400** to automatically install the driver obtained by the user.

It will be appreciated that, as used herein, a disposition may include more than one detected problem and suggested solution. For example, if a problem is detected with a printer **368** associated with a client computer **108**, the user may be provided with a disposition that offers the following suggested solutions: tightening a loose communication cable, replacing a toner cartridge, and updating the printer driver. Furthermore, if multiple problems are detected on a client computer **108**, the disposition returned to that client computer **108** may include a description of all of the detected problems, and a suggested solution or a source of additional information for each of the detected problems.

The operation of a client application **400**, in accordance with an embodiment of the present invention is depicted in **Fig. 6**. The initial state of the client application **400** after it has been downloaded to the client computer or device **108** is one in which it installs itself on the client computer or device **108** (step **600**). After the self-install procedure has been completed, the client application **400** proceeds to step **604** in which the identifier assigned by the server application **224** to the client computer **108** is obtained from the communications interface **404**. In general, the identifier is downloaded to the communications interface **404**, and is placed in the title bar of the communications interface **404**. In step **604**, the client application **400** reads the identifier from the title bar of the communications interface **404**. The identifier is thus provided to the client application **400** from the communications interface **404** over the identifier communication path **418**. According to this embodiment, apart from the communication of the identifier to the client application **400**, there is no direct communication between the client

application **400** and the client communications interface **404.** The described method for passing the identifier to the client application **400** is believed to be more reliable than other methods of passing this information, such as using "cookies," or methods that attempt to pass an identifier directly from the server application **224** to the client application **400**. In addition, it is believed to be more reliable than using the network address of the client computer **108** as an identifier available to both the client application **400** and the communications interface **404**. In particular, the described method ensures that the client application **400** and the communications interface **404** on a particular client computer or device **108** receive an identical identifier. The correct identifier is necessary in order to ensure that collected information and generated dispositions are associated with the same client computer or device **108**, as will be described more fully below. Of course, other methods of transferring an identical identifier to the client application **400** and the communications interface **404** can be employed in accordance with the present invention, so long as they are reliable in operation.

At step **608**, the client application **400** polls the server application **224** over a logical line of communication **412**. In particular, the client application **400** polls the server application **224** to detect a "run" command in a script status field contained in a record **420** containing user information associated with the client computer or device **108**. Upon detecting the value "run" in the script status field, the client application **400** downloads a next script **424** using logical line of communication **416** (step **612**). At step **616**, the client application **400** sets the script status field to a value "running." Next, at step **620**, the client application **400** runs the downloaded script **424**. The results obtained from running the script **424** are then returned to the server application **224** (step **624**). The script status field is then set to a value "done" by the client application **400**

(step **628**). Next, the value of the script status field is set to a value "ready" by the client application **400** (step **632**). The client application **400** then returns to step **608** to await detection of a next run command. In general, the next script **424** that is downloaded by the client application **400** is specified in a next script field included in the database record **420** containing user information. The specified script **424** is downloaded upon detection of the next "run" command in the script status field.

The operation of the communications interface **404** when used in connection with an embodiment of the present invention is depicted generally in **Fig. 7**. At step **700**, the main page is displayed. The main page provides information regarding the diagnostic service, and a form for collecting identifying and billing information concerning the user. After the user has completed the registration form, the registration information is returned to the server application **224** using the first line of communication **408** (step **704**). An identifier assigned by the server application **224** may then be downloaded to the communications interface **404** of the client computer **108** (step **708**). Alternatively, the identifier may be assigned immediately, and passed to the communications interface **404** with the registration form. It is desirable to provide the identifier to the communications interface **404** early on in the diagnostic process, so that any information collected regarding the client computer **108** and the user can be distinguished from information collected from other client computers **108** and users.

At step **712**, the communications interface **404** polls the script status field in the record **420** associated with the client computer **108**. Upon detecting a "done" status in the script status field, the communications interface **404** displays the next page (step **716**). The next page is indicated in a user section field of the record **420** containing user information. The next page

may simply be informative, or may also serve to collect information from the user. Where the next page only provides information, the communications interface **404** returns to step **712** to poll the server to detect a "done" status. Where the next page requires input from the user, the communications interface **404** will proceed to step **720**. For example, a displayed page may request authorization from the user before a script **424** is executed. After the information requested from the user has been returned to the server application **224** (step **724**), the script status field is set to "run" and the communications interface **404** returns to step **712**. As part of returning the client selection to the server application **224**, the communications interface **404** may set the script status field to "run", such as when the communications interface **404** has requested authorization from the user to execute a next script.

While the communications interface **404** waits for a "done" value to be detected, the previous page continues to be displayed (step **728**). For example, the communications interface **404** may display the message "working" while the client application **400** is executing a script **424**.

The polling procedures carried out by the client application **400** and the communications interface **404** allow the client application **400** and the communications interface **404** to be synchronized with one another. In particular, the client application **400** and the communications interface **404** use a field included as part of one of the records **420** maintained in the database **212** as a common point of contact. In this way, the operation of the client application **400** and the communications interface **404** is coordinated without requiring direct communication between the client application **400** and the communications interface **404**.

With reference now to **Fig. 8**, test procedures or commands that may be performed in diagnosing a client computer **108** are illustrated. Initially, an inventory of the client computer **108** devices and registry settings may be taken (step **800**). The inventory may be taken by an inventory module diagnostic tool comprising one or more scripts **424**. The inventory module may utilize several different methods for obtaining an inventory of the devices installed in the client computer **108** and the client computer's **108** registry settings. For instance, the inventory module may search a registry of installed devices and parse out from current and prior registry entries information concerning those devices. The inventory module may also obtain information from system utilities included as part of the operating system of the client computer **108**. Additionally, the inventory module may obtain information from the basic input/output system (BIOS) of the client computer **108.** In general, the inventory module may gather information from any file or software setting contained in the client computer **108** pertaining to devices currently or previously installed on the client computer **108**. Furthermore, the inventory module may collect information regarding general system resource and performance settings of the client computer **108**.

The inventory module may also collect information concerning the client computer **108** from the user. For example, the inventory module may include activating the communications interface **404** to ask the user to enter the name of the last program installed and the last program or file downloaded. The information gathered by the execution of the inventory module may be returned by the client application **400** to the server application **224** for storage in the database **212** as a part of a device inventory record **420** associated with the client computer using the identifier assigned to that client computer **108**. Alternatively, the information concerning inventoried

items is stored in a plurality of tables or records **420**. For instance, a first record **420** may be provided for storing information concerning a hard disk drive, a second record **420** may be provided for storing information regarding a video adaptor, a third record **420** may be provided for storing information regarding a CD ROM drive, etc. In general, records **420** may be provided for each desired division of information concerning devices and settings related to the client computer **108**.

At step **804**, functional tests of detected devices are performed. These functional tests may be implemented by executing a script **424** comprising a functional test module. The diagnostic tools run as part of the functional tests confirm that devices detected by the inventory module are functioning. For example, the functional test module may address each of the detected devices to request a response from those devices to ascertain whether those devices are in fact installed in the client computer **108**. The functional test module may also determine whether devices are operating properly. For example, the user may be instructed to insert media into devices such as floppy disk drives **340**, CD ROM drives **344**, tape drives **348**, and other devices having removable media, and to put paper in the printer **368**, etc. The functional test module may then test the ability of such devices to operate properly, such as to perform read and/or write operations, to print, etc.

At step **808**, performance or system data is collected. The performance data may include information on resource utilization, such as how much memory **332** is being used and how much space is available on the hard disk drive **340**. Performance data may also include the actual speed at which the processor **312** is running. The collection of performance data may be performed by a performance data module comprising one or more scripts **424**.

User input regarding the problem or problems to be diagnosed may be obtained at step **812**. According to one embodiment of the present invention, the user provides input by selecting from a drop down menu displayed by the communications interface **404**. For instance, the user may indicate the type of problem by choosing a menu item labeled "hardware," "software," "performance," or "don't know". Action taken in response to the selected user input will be described in greater detail below.

Regardless of the type of problem selected, any additional test applications or maintenance procedures that may be indicated are run at step **816**. Other test applications may comprise diagnostic tools that include specific utilities that may be downloaded to the client computer **108** when it is determined that such additional utilities or tests may be useful in diagnosing a problem on the client computer **108.** The log files created by the tests or maintenance programs are examined, and all or selected portions of the log files may be stored in the database as part of one or more records. Log files may be generated by executing functional tests and/or maintenance programs that may include but are not limited to diagnostic tools such as the "scan disk" and "disk defragmentation" utilities.

Although it is preferred that the describe inventory module, test modules and test applications be used, it should be appreciated that less than all may be used. For example, it may be appropriate to use only one test module during a particular session.

The results obtained from executing the various test modules and test applications are returned to the server at step **820**. According to alternative embodiments of the present invention, not all of the test results need be returned to the server as part of one batch of results.

For instance, results may be returned to the server application **224** following execution of each complete test module, or following execution of each individual script **424**.

At step **824**, a rules based analysis is applied to the test results to obtain a disposition. Finally, at step **828**, the disposition may be returned from the server application **224** to the client computer **108**.

With reference now to **Fig. 9**, the operation of an embodiment of the present invention in response to a user's indication that the problem to be diagnosed is a hardware problem is illustrated. Initially, at step **900**, the user selects the menu item relating to a hardware problem. At step **904**, the user selects from a list of devices to further specify the problematic hardware component. According to one embodiment of the present invention, the list of devices comprises the devices detected by the execution of the inventory module. According to a further embodiment of the present invention, the list of devices includes hardware devices in addition to the hardware devices detected by the inventory module, in order to allow the user to indicate a problem with a device that was not detected by the inventory module. According to still another embodiment of the present invention, the user may be allowed to specify a problematic device by entering the name of the problematic device or a description of that device.

After receiving the user information regarding the problematic device, additional tests on the device itself or on related software may be performed (step **908**). For example, additional test scripts may be downloaded to the client application **400** and executed. Also, device drivers, software versions, revision levels or hardware revisions associated with the identified device may be analyzed to determine whether patches or updates are available. In some cases, identifiable recall data or other pertinent manufacturer information may be relayed. The results of the

additional tests and/or software analysis is returned to the server application **224**, and a rules based analysis of those results is performed.

At step **912**, the disposition arrived at as a result of the tests and analysis of test results performed at step **908** is returned to the client computer **108**. In addition, the disposition obtained from the execution of the general test modules, as described in connection with **Fig. 8**, is returned to the client computer (step **916**).

From the above description of the operation of an embodiment of the present invention, it can be appreciated that allowing a user to specify a hardware problem is advantageous. In particular, it allows a disposition addressing the specified device to be returned to the user, even when the general test modules do not detect such a device. For instance, a disposition returned at step **912** might include a suggestion that the user verify the proper connection of power and data cables to a device identified by a user as problematic, but that was not detected by the general test modules.

As part of the general test performed regarding the client computer **108**, general system parameters and conditions may be analyzed. For example, the remaining space in the hard disk drive **336**, fragmentation of the hard disk drive **336**, the amount of hard disk drive **336** space occupied by discarded files, and the amount of memory **332** being used may be considered. Results regarding such general system parameters and conditions may be analyzed and a disposition returned to the client computer **108**. The dispositions returned may include information for the user, a suggestion to perform a maintenance procedure or system setting, or a script **424** may be provided to the client application **400** to automatically perform or run a maintenance procedure or to automatically change a system setting. The disposition may also

include a combination of disposition types. For example, the user may be provided with a disposition indicating that the hard disk drive **336** is fragmented, and a script may be provided to the client application **400** to automatically invoke a disk defragmentation utility already resident in the client computer **108**.

With reference now to **Fig. 10**, operation of an embodiment of the present invention in response to a user's selection of a software problem (step **1000**) is illustrated. At step **1004**, the user is asked to select the type of software problem that is being experienced. For example, according to one embodiment of the present invention, the user may select from the following choices:

1. My program gives me an error message before it shows on my screen;
2. My program starts, but when I try to use it, it gives me an error message;
3. My program runs, but it does not act right when I use it;
4. My program looks like it does not start at all; and
5. My program clears my screen to blue and gives me an error message.

Of course, additional or alternative menu choices may be provided. According to one embodiment of the present invention, additional menu items may be added as the number of problems having associated dispositions in the database increases. In general, after the user has specified the type of software problem, the user is instructed to start the problematic program after the communications interface **404** closes (step **1008**). At step **1012**, the communications interface **404** is closed, while the client application **400** continues to run on the client computer **108**. At step **1016**, a timer is started in order to ascertain how much time is required for the client computer **108** to load the problematic program. At step **1020**, it is determined whether the timer

has expired. If the timer has expired, the user is reminded to start the program if the user has not already attempted to do so (step **1024**).

At step **1028**, it is determined whether an error message has been generated in connection with the user's attempt to start the problematic program. If an error message has been generated, the text of the message is parsed and returned to the server application **224** for a rules based analysis of that message (step **1032**).

At step **1036**, it is determined whether the program is frozen or will not load. If the program is frozen or will not load, the point at which execution failed is determined (step **1040**). In particular, the executable program, .dll file, or other file causing the software to hang-up or fail to load is determined. The information regarding the point at which execution failed is returned to the server application **224** for rules based analysis.

The disposition obtained as a result of the rules based analysis or analyses conducted as a result of the user's selection of a software problem is then returned to the client computer (step **1044**). In addition, the disposition obtained from a rules based analysis of the general test results is also returned to the client computer **1008** (step **1048**).

From the above description of the operation of an embodiment of the present invention, it can be appreciated that the present invention is capable of precisely identifying the nature of a software problem. In particular, the present invention can determine whether a problem with software is due to the operating system, a conflict with other programs or shared components or the problematic software itself. This is achieved without requiring the user to do anything more than attempt to load the problematic program. In addition, it does not require a trained technician

to analyze the problem. Instead, the results of the various tests are returned to the server application **224** and a rules based analysis of those results is performed by the server **104**.

Fig. 11 illustrates the operation of an embodiment of the present invention in response to the selection of a performance problem by the user (step **1100**). At step **1104**, the user selects the particular type of performance problem that is being experienced. For instance, according to an embodiment of the present invention, the user may select from the following general types of performance problems:

1. My computer's performance in general is slow;
2. My computer has difficulty opening any selected program; and
3. My computer has a problem with a particular program or window.

At step **1108**, it is determined whether the user has selected a problem with performance in general. If a general performance problem has been selected, a rules based analysis of the system settings is performed (step **1112**). As will be appreciated by those of ordinary skill in the art, system settings may include the speed of the processor **312**, the amount of free RAM **332**, space allocated to virtual RAM on the hard disk drive **336**, etc. In addition, a rules based analysis of the time required to open an application commonly found on client computers **108**, the number of files on the hard disk drive **336**, and the number of programs always running in the background may be performed.

At step **1116**, it is determined whether a user has indicated a problem with the opening or operation of any selected program. If a problem with the opening or operation of any selected program has been selected, a rules based analysis of system settings (step **1112**) is performed. In addition, a check for conflicts may be performed.

At step **1120**, it is determined whether the user has selected a problem with a specific program or window. If yes, the software problem analysis procedure described above with respect to **Fig. 10** is performed (step **1124**). As described with respect to **Fig. 10**, the software problem analysis may include a rules based analysis of information returned to the server application **224** as a result of performing the software problem analysis.

From the description set forth above it can be appreciated that, in general, information concerning the type of problem specified is returned to the server application **224** and a rules based analysis of that information is performed. The analysis may further comprise an adaptive rules based analysis. After application of the rules based analysis of the returned information, regardless of the particular performance problem specified, a disposition is returned to the client computer **108** (step **1128**). In addition, at step **1132**, the disposition obtained from performance of the general test modules is returned to the client computer **108**.

From the description set forth above of an embodiment of the present invention, it can also be appreciated that the specification of a performance problem may result in the performance of tests in addition to those performed during execution of the general test modules. In addition, from the example given above, it is evident that the procedure otherwise carried out in response to the specification of a software problem may also be performed in response to the specification of a performance problem. The above-described configuration of an embodiment of the present invention enables a user who perceives a performance problem on the client computer **108** to obtain a diagnosis of the problem without needing to identify the nature of that problem with particularity. Accordingly, the present invention may allow a user to identify and fix a performance problem even if the user is not an experienced computer user.

Fig. 12 illustrates the operation of an embodiment of the present invention in response to a selection by the user indicating that the user does not know the nature of the problem (step **1200**). In particular, a menu selection to indicate that the user does not know the nature of the problem allows a diagnosis to proceed even if the user is unable to ascertain whether the particular problem involves the hardware, software, or performance of the client computer **108**.

In response to the selection of "don't know" as the type of problem in step **1200**, according to one embodiment of the present invention, certain of the tests and analyses conducted in response to the selection of a performance problem may be conducted. In particular, the system settings and operating system settings may be subjected to a rules based analysis (step **1204**). In general, any test procedure associated with any specified problem type may be conducted in response to the selection of "don't know" at step **1200**. In particular, any test procedure that does not require the user to take any additional action, such as specifying a hardware device or software program causing problems, may be performed.

At step **1208**, a disposition obtained from the specific and general tests performed concerning the client computer **108** is returned to the client computer (step **1208**). In addition, information may be returned to the client computer **108** regarding common computer problems and how to fix such problems.

From the description of the operation of an embodiment of the present invention in response to an indication by the user that the specific type of problem is not known, it can be appreciated that any test can be performed on the client computer **108**, so long as specific information from the user is not required. Of course, according to other embodiments of the present invention, tests that do not require additional input from the user may be performed

regardless of the input regarding the particular type of problem selected by the user. The performance of such tests may provide a more accurate diagnosis of the problem or problems on the client computer **108**. However, it may be desirable to limit the number of tests that are performed automatically, such as where the user has opted for only a basic diagnostic service, or where it is important to limit the time required to return a diagnosis.

According to another embodiment of the present invention, in response to a selection of "don't know", no additional tests are performed. Results obtained from test modules that have been executed are then subjected to a rules based analysis. In this way, any test results that indicate or that may indicate a problem with the client computer **108** can be identified. A disposition may be returned to the client computer **108** with respect to each identified problem.

According to still another embodiment of the present invention, in response to the selection of "don't know", the user may be presented with an additional question or questions, in an attempt to more particularly identify the problem. For instance, the user may be asked under what circumstances problems are encountered. Based on the user's answers, areas of potential problems can be identified. For example, such questions may be successful in narrowing the potential problem to a hardware problem, and test results concerning devices associated with the client computer **108** can be subjected to a rules based analysis.

With reference now to **Fig. 13**, the operation of an embodiment of the present invention in returning a disposition to the client computer **108** is illustrated. Initially, at step **1300**, the results returned to the server application **224** as a result of the performance of tests on and gathering of information from the client computer **108** are entered into records **420**. In general, at least one record **420** associated with the client computer **108** from which the results were

obtained is created in the database **212**. According to an alternative embodiment of the present invention, a plurality of records **420** associated with a particular client computer **108** are created. The creation of multiple records allows the results to be categorized, thereby simplifying the analysis and retrieval of stored information. Multiple records in a single category may be created where, for example, there are multiple devices in that category. For instance, a record **420** in the category of "storage devices" may be provided for a first hard disk drive **336**, a floppy drive **340**, and a second hard disk drive **336** detected in a single client computer **108**. In general, all of the records **420** associated with a particular client computer **108** include a field containing the identifier assigned to that client computer **108**, so that the stored information can be associated with the correct client computer **108**.

At step **1304**, a rules based analysis of the information stored in the database records **420** is performed. According to one embodiment of the present invention, the information is analyzed as discrete pieces or sets. For example, a rules based analysis may first be performed concerning a hard disk drive **336** installed on the client computer **108**. More particularly, the rules based analysis may obtain information from a record or records **420** containing only information concerning a hard drive **336**, or only information concerning the hard drive **336** may be taken from a record or records **420** containing other information, for analysis. A second rules based analysis may then be performed concerning another detected device or aspect of the client computer **108,** such as a CD ROM **344** or printer **368**. In general, it can be appreciated that a series of rules based analyses may be performed as required or indicated by the detected devices, software and parameters concerning the client computer **108**, or as indicated by the type of problem specified by the user.

As described more fully above, the rules based analysis of information may comprise reviewing the collected information to determine whether an anomaly or potential conflict is present.

At step **1308**, it is determined whether a potential problem is indicated as a result of the rules based analysis. If a potential problem is found, any potential dispositions associated with the indicated problem are stored (step **1312**). For example, a record **420** may be provided for storing any such potential dispositions. As with other records **420** associated with a particular client computer **108**, the record or records **420** may be associated with the client computer **108** to which the disposition pertains by assigning the identifier used to identify the client computer **108** to the record or records **420**.

At step **1316**, it is determined whether any additional test results or information returned from the client computer **108** require analysis. Accordingly, where the returned results are analyzed as discrete sets of information, the system will cycle through steps **1304**, **1308**, **1316** and if a problem is found, step **1312**, until no results or information requiring analysis remain. The potential disposition or dispositions are returned to the client computer **108** at step **1320**.

The disposition returned to the client computer **108** may comprise a suggested action to remedy a detected problem, may provide information regarding a source of further information regarding the detected or potential problem, or may send the user to a site containing relevant information. For instance, a returned disposition may comprise a suggestion that the user execute a particular utility program or perform a physical check of a device. In addition or as an alternative to suggesting a particular procedure, the disposition may provide information, such as an Internet address (*e.g.*, a Universal Resource Locator) or telephone number of a source of

further information, such as the manufacturer of a problem device or program. The disposition may also comprise an inventory of installed devices or software, and may include information regarding the status of the devices or software.

The disposition may also include provisions to automatically initiate remedial action. For example, the user may be asked to authorize the performance of the suggested maintenance procedure. Upon receiving the requested authorization, a script **424** may be provided to the client application **400** for execution. The script **424** may itself comprise a utility application, or may comprise the programming code necessary to initiate operation of a utility program stored on the client computer **108**. In either event, the procedure may be performed without requiring the user to do anything more than authorize execution of the procedure. It should be noted that scripts **424** run during the diagnostic procedure prior to the return of a disposition to the client computer **108** may have remedied the problem or problems that caused the user to seek a diagnosis and solution. For instance, according to one embodiment of the present invention, disk maintenance utilities provided as part of a script or scripts **424** may automatically be run during diagnosis of the client computer **108**. These utilities may be effective in remedying the problem with the client computer **108**, without requiring further repair. In such instances, the disposition returned to the client computer **108** may comprise notice to the user that a problem with a component (e.g. the hard disk drive **336**) was detected, but was fixed when a provided disk maintenance utility was executed (e.g., SCAN DISK). The utilities may be generally available programs, such as SCAN DISK, or they may be specifically developed for use in connection with the present invention.

If a problem with a particular piece of software is detected, the disposition may include the provision of a software patch, missing component, or replacement of the software to remedy the problem. Accordingly, the disposition may be provided with a request that the user authorize replacement of the faulty software. In response to receiving such authorization, the software patch or replacement may be downloaded to the client computer **108** for installation. The software patch or replacement may be automatically installed by the client application **400**. If the required software patch or replacement is not available from the server **104**, the communications interface **404** may be directed to a website where the required software is available. The user may then download the required software to the client computer **108**. According to one embodiment of the present invention, the downloaded software may be automatically installed by the client application **404**.

It should be appreciated that multiple dispositions may be returned to the client computer **108**. For example, a suggestion to perform a maintenance procedure to improve the performance of a hard disk drive **336** installed on the client computer **108** as a first disposition may be accompanied by a suggestion to replace a device driver for that disk drive **336**. In addition, dispositions may concern more than one device or piece of software associated with a client computer **108.**

The operation of an embodiment of the present invention will now be illustrated in the context of an example. Initially, the user contacts the server **104** over the computer network **112** using the communications interface **404** of the client computer **108**, such as an Internet browser (step **500**). The user is presented with a browser page describing the diagnostic service. If the user decides to continue, a registration form is presented to the user by the communications

interface **404**. The registration form or page allows the user to submit general and billing information. For example, the user is asked to enter his name, address, and billing information such as a credit card number and card expiration date. The user may choose to submit the registration form after entries have been made in all of the required fields of that form. The information submitted is stored in a record **420** containing user information.

The record **420** containing user information includes a field for storing an identifier that is unique to the particular client computer **108** that will be diagnosed. The identifier is assigned by the server application **224** when the user submits information from the registration form. In general, all communications and records **420** with respect to a particular client computer **108** are identified by the same identifier. In this way, information passed between the server **104** and the client computer **108**, or stored in the database **212**, can be properly associated with the correct client computer **108**.

Following registration, the communications interface **404** displays a next page to the user. The signal to display a next page is communicated to the communications interface **404** by setting a script status field included in the record **420** containing user information to a value "done" upon completion of the collection and/or verification of registration information. The identity or address of the next page to display is contained in a next page field included in the record **420** containing user information. In general, the communications interface **404** may detect a change to the script status field by polling the server application **224** (step **712**). The next page displayed (step **716**) is, according to the present example, a page informing the user that components of the client executable package **220** will next be downloaded and installed on the client computer **108**, and request authorization from the user to commence with the download

and installation. Upon receiving the required authorization, here signaled by having the user click on a button displayed by the communications interface **404**, the client application **400** is downloaded to and installed on the client computer **108** (step **504**).

After it has been downloaded, the client application **400** self installs itself on the client computer **108** (step **600**). In order to ensure that information returned to the server application **224** is associated with the relevant client computer **104**, the client application **400** must be provided with the identifier that has been assigned to the client computer **108**. According to the present example, the identifier is provided to the communications interface **404**, and displayed in the title bar of the communications interface **404** window. Thus, upon completing installation, the client application **400** reads the identifier from the title bar of the communications interface **404**. In this way, the identifier is reliably communicated to the client application **400**. According to the embodiment of the present example, the reading of the identifier from the title bar of the communications interface **404** is the only direct communication between the client application **400** and the communications interface **404**.

The client application **400** then polls the server application **224**. In particular, the server application **400** polls the server application **224** to detect a "run" value in the script status field of the record **420** containing user information. Because the client application **400** has been provided with the identifier for the client computer **108** on which it is installed, the client application **400** monitors only the record **420** having user information concerning the relevant client computer **108** (step **608**). Upon detection of the "run" command, the client application **400** downloads an application script **424** (step **612**). After the download of the script **424** is

complete, the client application **400** sets the script status field in the record **420** containing user information to "running" (step **616**) and executes that script (step **620**).

According to the present example, the first script **424** conducts an inventory of devices installed on the client computer **108** and the registry settings in the client computer **108**. In general, the information regarding installed devices and registry settings may be collected by queries made to the operating system of the client computer **108**, such as by querying standard system settings and device information, and by extracting information from files concerning installed devices and registry information using programming designed to parse such information from such files. Additional information may be obtained by querying the system basic input/output system (BIOS). Execution of the device inventory and registry setting module may comprise the execution of one or a plurality of scripts **424**.

Following the execution of the device inventory and registry settings test module, the results obtained are returned to the server application **224**. The server application **224** stores the returned information in database records **420**. According to the present example, a separate database record **420** is provided for storing information related to various devices, manufacturers, and system settings. For example, a separate database record **420** may be provided for each of the following categories of information: monitor information; port information; printer information; storage device information; system information; video adaptor information; performance information; software information; and audio information.

Multiple records **420** of a particular type may be provided in connection with a particular client computer **108**. For instance, a client computer **108** having three active communication ports will have three records **320** for containing port information.

Following execution and the return of information to the server application **224**, the client application **400** sets the script status field in the record **420** containing user information to "done" (step **628**). The communications interface **404** detects the change of the script status field to "done" because it has been polling that field (step **712**) while the client application **400** has been executing the script **424**. The communications interface **404** then displays a next page (step **716**) to the user. According to the present invention, a plurality of scripts **424** comprise the device inventory and registry settings test module. Accordingly, the server application **224** may change the script status field to "run" immediately after the client **400** has indicated that it is ready to execute a next script **424** by changing the script status field to "ready" (step **632**). Thus, the process of downloading a script **424** and executing that script **424** continues until all of the scripts **424** needed to be run in connection with the device inventory and registry settings module have been executed.

Next, functional tests are performed on the detected devices. According to the present example, the client computer **108** includes a hard disk drive **336**. Accordingly, one of the functional tests that will be performed includes a hard disk drive diagnostic program. In addition, the client computer **108** according to the present example includes a floppy disk drive **340**. Accordingly, one of the functional tests that will be carried out on this client computer **108** includes a test of the operation of the floppy disk drive **340**. Because a more complete test can be performed on storage devices having removable media, such as floppy disk drives **340**, if the media is installed, the user is asked to insert media into all installed drives at step **716**. After the user has done so, the user signals this to the server application **224** by clicking a button displayed by the communications interface **404** to indicate that functional testing may commence (step

720). The user's indication is returned to the server application **224** (step **724**), at which point the script status field is changed to "run."

As described above, the client application **400** detects the "run" value in the script status field, downloads the next script **424** (step **612**), sets the script status field to "running" (step **616**), runs the script (step **620**), returns results from the execution of the script to the server application **424** (step **624**), sets the script status field to "done" (step **628**), and then changes the script status field to "ready" (step **632**). The information collected by execution of the scripts that comprise the functional test module is, according to the present example, returned to the server application **224** after the execution of each script **424**. The information is stored in the records **420** created for the previously detected devices and registry settings. Accordingly, the information returned to the server **104** as the result of the execution of the functional test module may be stored as entries in records **420** that have previously been created.

According to the present example, performance data is next collected. The execution of the scripts **424** comprising the performance data test module, and the return of information thus collected to the server **104** proceeds essentially as described above with respect to the execution of the device inventory and registry settings module and the functional test module. As before, the performance data information may be stored in one or more records **420** that were created earlier for storing information concerning the client computer **108**.

An inventory is displayed of all devices and settings as well as software executables (.exe). The user is next asked to provide information regarding what the user perceives to be the problem requiring diagnosis (step **812**). For example, the user may be asked which program last loaded, last Internet download, etc. According to the present example, the user indicates that the

client computer **108** has a hardware problem (step **900**). The user is then presented with a list of devices detected in the client computer **108** (step **904**). The user is also presented with a list of devices typically found in a client computer **108**. For purposes of the present example, we will assume that the user indicates that the CD ROM drive **344** is problematic. We will further assume that the CD ROM drive **344** was not detected by the device inventory and registry settings test module. Because the CD ROM **344** was not detected, no additional tests are performed relating to such a device (step **816** and **908**).

Next, any information concerning the client computer **108** remaining to be returned to the server **104** is provided to the server application **224** (steps **820** and **912**).

The server application **224** then applies a rules based analysis to the information stored in the records **420** concerning the client computer **108**. For purposes of the present example, we will assume that no problems regarding devices detected as being installed or detected registry settings are considered to be problematic. However, the user's indication of a problem with a CD ROM drive **344** does require diagnosis. According to the present example, no CD ROM **344** was detected as installed on the client computer **108**. This suggests that the CD ROM **344** is not properly connected to the client computer. Therefore, the disposition returned to the client computer **108** (steps **828, 912**, and **916**) may suggest that the user ensure that the signal and power cables between the client computer **108** and the CD ROM **344** are intact and properly connected.

From the above example, it can be appreciated that the user was provided with a diagnosis of the problem on the client computer **108**, without requiring the assistance of a human technician. Furthermore, although in the present example the user provided information

concerning the perceived problem, the user was not required to describe the problem with any particularity. In addition, the fully automated test procedure, in combination with the user input, provides a more narrowly focused disposition than might otherwise be available. For instance, without the automatic device inventory and review of system settings, suggested remedial measures would necessarily be broader. For instance, a system that did not prepare an automatic inventory might instruct the user to update the driver for the CD ROM **344**, or ask the manufacturer of the drive for assistance. However, the present invention, as illustrated in the foregoing example, is capable of providing a narrowly focused suggestion for remedying the problem. Accordingly, the present invention is capable of streamlining the process of diagnosing and remedying problems on a client computer **108**.

The operation of the present invention will now be described in the context of a second example. According to this second example, the operation of the present invention proceeds as described above in the previous example. However, according to this example, the user selects a performance problem (steps **812** and **1100**). Next, the user selects the particular type of performance problem being experienced (step **1104**). According to the present example, the user indicates that there is a problem with a particular program (step **1120**).

Because the user has indicated a problem with a particular program, a software performance analysis is begun (step **1128**). In particular, the user is instructed to attempt to start the problematic program after the communications interface **404** has closed (step **1008**). Next, the communications interface **404** automatically closes (step **1012**) and a timer is started (step **1016**).

The user then clicks on the problematic program or otherwise attempts to load it. According to the present example, we will assume that the program freezes while it is loading (step **1036**). At step **1040**, it is determined at which point loading of the program failed. In the present example, it is determined that the program freezes when a particular file is being loaded. Information concerning this file is returned to the server application **224** and a rules based analysis is performed (step **824**).

The disposition returned to the client computer **108** (step **828** and **1044**) may inform the user of the name of the file or component being loaded when the program freezes, and may request authorization from the user to replace that file or component. The application server **224** may then retrieve a copy of the affected file, for example, from a website operated by the software producer. The system **100** may then automatically install the patch or replacement file in the client computer **108**.

As illustrated by this second example, the present invention is capable of precisely determining the problem being experienced by a user in connection with a particular piece of software. Furthermore, the present invention is capable of providing the user information regarding the problem, and of remedying the problem. It is further apparent that the present invention is capable of diagnosing a problem even where the user is unable to describe the problem with specificity.

In accordance with the present invention, a method and apparatus for diagnosing computer hardware and software are provided. The invention in its broader aspects relates to a method and apparatus for diagnosing computer hardware and software without requiring input from a user or technician who is experienced in computer repair. In particular, the present

invention provides a method and apparatus that are capable of diagnosing computer hardware and software while receiving only basic input from a user of a computer to be diagnosed, and without requiring human intervention at the server side of the system. Furthermore, the method and apparatus of the present invention is capable of providing repair information, including information necessary to effect a repair automatically.

Although the present invention includes descriptions of particular embodiments, it will be appreciated that various modifications are possible and are within the scope of the description. For example, menu items that may be selected by the user to describe a perceived problem or problems may include items in addition or as alternatives to those described in the foregoing description. As a further example, the computer to be diagnosed may be part of a local computer network, rather than a stand alone computer. Furthermore, the present invention may be used to diagnose digital devices other than personal computers, including personal digital assistants, wireless devices of various configurations, or other devices capable of communicating with a diagnosing server.

The foregoing discussion of the invention has been presented for purposes of illustration and description. Further, the description is not intended to limit the invention to the form disclosed herein. Consequently, variations and modifications commensurate with the above teachings, within the skill and knowledge of the relevant art, are within the scope of the present invention. The embodiments described hereinabove are further intended to explain the best mode presently known of practicing the invention and to enable others skilled in the art to utilize the invention in such or in other embodiments and with various modifications required by the

particular application or use of the invention. It is intended that the appended claims be construed to include alternative embodiments to the extent permitted by the prior art.